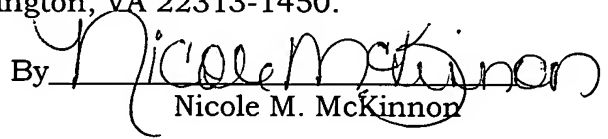


**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

APPLICANT: Christian Prehofer, et al. GROUP: Not Yet Assigned  
SERIAL NO.: Not Yet Assigned EXAMINER: Not Yet Assigned  
FILED: HEREWITH  
FOR: RECONFIGURATION OF A GROUP OF NETWORK NODES IN A  
WIRELESS AD-HOC NETWORK

.....  
**CERTIFICATE OF EXPRESS MAILING (Label No.: EV 492340539 US)**

I hereby certify that this correspondence is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. Section 1.10, on February 22, 2005 and is addressed to Mail Stop PCT, Commissioner for Patents, P.O. Box 1450, Arlington, VA 22313-1450.

By   
Nicole M. McKinnon

.....  
Mail Stop PCT  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**SUBMISSION OF SUBSTITUTE SPECIFICATION**

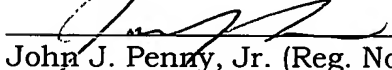
Sir:

The Substitute Specification includes the amendments that were made under PCT Article 34.

Applicant submits that no new matter has been added to the invention in the substitute specification.

Respectfully submitted,

Date: February 22, 2005  
Customer No.: 21874  
478864

By:   
John J. Penny, Jr. (Reg. No. 36,984)  
Intellectual Property Practice Group  
EDWARDS & ANGELL, LLP  
P.O. Box 55874  
Boston, MA 02205-5874  
Tel.: (617) 517-5549  
Fax: (617) 439-4170

CLEAN COPY OF SUBSTITUTE SPECIFICATION

*Reconfiguration of a  
Group of Network Nodes  
in an Ad-hoc Network*

**FIELD OF INVENTION**

The present invention relates to a reconfiguration of a group of network nodes in an ad-hoc network, and in particular to a reconfiguration of a group of network nodes in an ad-hoc network with particular emphasis on reconfiguration consistency.

**BACKGROUND ART**

In WO 01/14968 A1 there is described a fieldbus upgradable apparatus and method, wherein control devices residing on a fieldbus communications network are modified without interrupting the operation of the control devices in a seamless manner.

In US-A-6,113,652 there is described a communication network equipment capable of non-disruptive software upgrade which is used in a telecommunications network having a plurality of

coupled nodes and a network controller coupled to at least one of these nodes.

Further, in US-6,141,683 A there is described a method for  
5 remotely and reliably updating the software on a computer  
with provision for roll-back with particular focus on  
integrity between different software versions.

Further, in US-A-5,699,275 there is described a system and  
10 method for remote patching of operating code located in a  
mobile unit, wherein a manager host is operable to initiate  
transmission through a communication network of at least  
one discrete patch message defining at least one patch.

15 Further, in US-2001/0029178 A1 there is described a  
wireless software update with version control in a wireless  
communication system having a system backbone, having a  
host computer coupled to the system backbone, at least one  
base station coupled to the system backbone, and a  
20 plurality of mobile devices within the system.

Further, in WO 01/84792 A1 there is described a method and  
gateway for performing an online switching of software in a  
communication system.

25

Overall, numerous factors associated with technology,  
business, regulation and social behavior have driven the  
spreading of wireless ad-hoc networking in the past, i.e.,  
a wireless network formed without any central  
30 administration. Ad-hoc networks consist of a plurality of  
mobile devices using a wireless interface for exchange of

packet data. As each mobile device in the ad-hoc network serves as router and host, each such mobile device will forward data packets on behalf of other mobile devices and further run user applications. Therefore, in ad-hoc  
5 networks, mobile devices are connected directly for local cooperation.

To support on-going improvement of functionality, an ad-hoc networks should provide the opportunity for a software  
10 update. It is often required that all mobile devices in such an ad-hoc network have the same software version, e.g., for reasons of compatibility. For this reason, the software update should take place in a coordinated manner, preferably at the same point in time. In addition, either  
15 all mobile devices reconfigure successfully, or they all fall back to the software version before installation.

However, up to now, no appropriate approach to the coordinated update of software in ad-hoc networks is  
20 available. Another issue not addressed so far is that during reconfiguration, mobile devices may not communicate properly.

## 25 **SUMMARY OF INVENTION**

In view of the above, the object of the present invention is to provide a solution to consistent software reconfiguration in ad-hoc networks.  
30

According to the present invention, this object is achieved through a method of reconfiguration for a network node, e.g., a mobile device or a stationary device, in an ad-hoc network. The method comprises a first step of preparing a transition from an initial software configuration to a target software configuration and a second step to deciding on commitment to the target software configuration. According to the present invention, the decision on commitment is taken in view of a result of reconfiguration indicated through at least one further network node in the ad-hoc network, in particular when every result of configuration received at the network node from a reachable further network node is evaluated to be positive.

Therefore, according to the present invention, a transition from an initial software configuration to a target software configuration is not executed anyway but such a transition is taken on the basis of information being related to reconfiguration at network nodes being reachable from the network node deciding on commitment to the target software configuration.

In particular, if the information received is related to further reconfiguration processes in the further network nodes, it is possible to allow for coordination of reconfiguration between different network nodes, although each single network node is operating autonomously.

In other words, according to the present invention, it is proposed to not simply initiate an update process for different network nodes in an ad-hoc network and to just

hope that the reconfiguration will be successful, but to use feedback mechanisms locally in each single network node to decide on commitment to target software configurations.

5 According to a preferred embodiment of the invention it is proposed to negotiate a maximum reconfiguration time period between at least two network nodes executing reconfiguration. The maximum reconfiguration time is the maximum time for reconfiguration, indication of  
10 reconfiguration result, and executing a fallback to the initial software configuration for network nodes in the ad-hoc network participating in the reconfiguration process.

15 Further, preferably, a start of reconfiguration is coordinated between at least two network nodes in the ad-hoc network executing a reconfiguration process. Optionally, one network node in the ad-hoc network may organize the negotiation process.

20

The negotiation of a maximum time for reconfiguration and the "synchronization" or coordination of start of reconfiguration are of particular relevance for achieving a simple, but nevertheless highly efficient framework of  
25 autonomous reconfiguration in each network node.

The negotiation process of a maximum reconfiguration time period is adapted to a heterogeneous update of software within different network nodes, as the maximum time period  
30 for reconfiguration will be set according to the target

software configuration in the network node requiring the longest time period for reconfiguration.

Yet another advantage is that a time period during which  
5 services are interrupted due to reconfiguration at the different network nodes is foreseeable and minimized. If after expiry of the negotiated maximum time period no definitive reconfiguration success is established in the ad-hoc network, immediately counter-measures may be taken  
10 to bring the ad-hoc network back to operation on the basis of the initial software configuration before start of the reconfiguration process.

According to a further preferred embodiment of the  
15 invention, it is proposed to determine which network nodes are reachable after the reconfiguration process.

This preferred embodiment of the present invention is provided to control two different reconfiguration  
20 scenarios.

A first scenario would be that reconfiguration is not related to communication software - e.g., like speech-codec software - but to some type of application. Therefore,  
25 during reconfiguration communication is still possible. In such a case, connectivity from the network node executing the reconfiguration process to further network nodes running the reconfiguration process is not an issue.

30 A second scenario would be that some type of communication-related software is reconfigured so that it may not be

guaranteed that the ad-hoc network topology remains unchanged during the course of reconfiguration. One example would be a communication software update failure at a specific network node which leaves the ad-hoc network as it  
5 may no longer communicate with those network nodes which successfully reconfigured communication software. Therefore, excluding this network node from the commitment decision - after determination of reachability - allows to avoid over-restrictive commitment decisions and a related  
10 decrease in functionality improvement.

Further preferred embodiments of the present invention relate to the decision mechanisms underlying the commitment to a new software configuration in a network node.

15

One option is to commit to the target software when every result of reconfiguration received at the network node is positive. This type of commitment achieves maximum consistency of software versions after reconfiguration as  
20 only a fully successful reconfiguration process at different network nodes allows for transfer to the target software configuration in each single network node.

A further option relates to the other extreme case where no  
25 information regarding the outcome of reconfiguration at other network nodes is received at all. In this case - after expiry of the maximum reconfiguration time period - it is proposed to execute a fallback to the initial software configuration. The rationale behind this approach  
30 is that, initially, communication was possible on the basis of the initial software configuration. Further, the fact



that no reconfiguration result was received at the network node deciding on the commitment to the new software configuration is an indication that communication on the basis of the newly installed software is not possible.  
5 Therefore, fallback to the initial software configuration is the optimal way to re-establish connectivity in the ad-hoc network.

Yet another option is to execute a fallback to initial  
10 software configuration, when at least one negative reconfiguration result was reported to the network node deciding on commitment to the target software configuration. The rationale behind this commitment decision is to maximize connectivity and communication  
15 capability. Assuming that only a single mobile network node could leave the ad-hoc network according to the negative reconfiguration result, it is wise to continue the operation on the basis of the initial software configuration to maintain maximum connectivity.

20

Further preferred embodiments of the present invention relate to the involvement of a network node into the exchange of reconfiguration results.

25 According to a preferred embodiment of the present invention, each network node either sends a positive or negative indication of reconfiguration, according to the outcome of the transition from the initial software configuration to the target software configuration.

30

It should be noted that the exchange of reconfiguration results may be achieved in various ways.

5 A first way would be to indicate the reconfiguration result through dedicated positive or negative signals. Preferably, these signals may be processed, both, by through the initial software configuration and the target software configuration.

10 A second way would be to indicate a positive reconfiguration, e.g., of communication software, implicitly through automatic set-up of network connectivity.

15 For both alternatives given above, the exchange of reconfiguration results may be executed repeatedly to improve penetration of reconfiguration results in the ad-hoc network.

20 A further preferred embodiment of the present invention relates to the determination of network nodes in the ad-hoc network participating in the reconfiguration process.

25 According to the present invention, it is proposed to use a plurality of criteria for such determination which may be used either alone or in combination.

A first such criterion is communication capability of a network node. Here, it is proposed to have a match between  
30 the target software configuration and the related communication capabilities of a network node. In other

words, it would be meaningless to install communication software at a network node which is not applicable in view of the hardware support given by the network node. Further, communication capability may also be understood in the sense that only those network nodes are considered during software reconfiguration which support a specific style of communication. A further criterion is network connectivity, which means that network nodes not being reachable in the ad-hoc network are per se excluded from the reconfiguration process.

Yet another criterion may be profile data for a further detailed specification of the network node or its end user. Such profile data may classify whether a network node supports specific types of services which form part of the reconfiguration process. Using such profile data it is possible to avoid unnecessary retrieval of software and related data traffic in the ad-hoc network.

Yet another criterion is a movement pattern of a movable network node, i.e. a mobile device, which may either be predicted or be known per se. Assuming that - in view of the movement pattern - a mobile device will soon leave the ad-hoc network, it is highly meaningful to avoid unnecessary data traffic to such a mobile device for reconfiguration with respect to services being of importance only within the established ad-hoc network.

Yet another criterion that might be used as basis for determining network nodes participating in the reconfiguration in an ad-hoc network is a status of a

network node, typically a hardware status. One such example would be the energy that is available within a network node, which is of particular relevance for mobile devices. Here, when the amount of energy is not sufficient to support a complete reconfiguration process, it is reasonable to exclude the network node from the reconfiguration process to avoid any disturbance of reconfiguration for the remaining network nodes in the ad-hoc network. Another example of a hardware status may be the type of standard being supported for circuit-switched or packet-switched communication through the network node or processing power provided through processor devices in the network node. The reconfiguration of services only makes sense when a network node offers a certain processing power required for certain services.

Yet another example of a criterion to determine which network nodes should be reconfigured could be a group membership of a network node, which group membership may be specified in advance. This criterion is of particular value for supporting different user groups or closed user groups, e.g., in police departments or hospitals. Another grouping may be according to companies, membership to specific organizations, profession of end user, etc.

Yet another criterion for determination of network nodes participating in a reconfiguration could be a priority assigned to the end user of a network node. This criterion facilitates the ranking of end users, which is of benefit for providers of commercial services aiming at user selectivity.

It should be noted that the determination of network nodes as outlined above could either be executed before start of reconfiguration or be repeated during the reconfiguration process. The latter option allows to identify additional network nodes dropping out of the reconfiguration process, and therefore to minimize traffic in the ad-hoc network and to accelerate the reconfiguration process.

Another preferred embodiment of the present invention relates to retrieval of software for executing the transition from the initial software configuration to the target software configuration.

According to a preferred embodiment, software may be retrieved locally from a portable electronic device, e.g., a smart card or any other type of portable device that allows transfer of the related data.

Yet another option is to retrieve software via a mobile communication environment of any type, e.g., from the Internet via a personal area network, a wireless local area network, a wireless infrared communication network, Bluetooth communication, or any other suitable type of mobile communication environment.

Further preferred embodiments of the present invention relate to the type of software which is installed for reconfiguration.

As should be understood from the above, the present invention is not restricted to any type of software. The present invention may be used to reconfigure application software, communication software, operating system  
5 software, firmware, etc.

Further, the inventive reconfiguration process is not only related to software as such, but also to related control parameters. Therefore, one might imagine a situation where  
10 the software as such remains unchanged and only the parameter controlling the operation of the software are reconfigured in the sense outlined above.

According to the present invention, also heterogeneous  
15 software reconfiguration using network node specific software for transition to the target software configuration is fully supported. Therefore, the present invention is perfectly suited to support software reconfiguration with utmost flexibility and user specific  
20 adaptation.

According to another preferred embodiment of the present invention there is provided a computer program product directly loadable into the internal memory of a network  
25 node of an ad-hoc network comprising software code portions for performing the inventive reconfiguration process when the product is run on a processor of the network node.

Therefore, the present invention is also provided to  
30 achieve an implementation of the inventive method steps on computer or processor systems. In conclusion, such

implementation leads to the provision of computer program products for use with a computer system or more specifically a processor comprised in, e.g., a mobile device like a mobile telephone, a laptop computer, or a  
5 PDA.

These programs defining the functions of the present invention can be delivered to a computer/processor in many forms, including, but not limited to information  
10 permanently stored on non-writable storage media, e.g., read only memory devices such as ROM or CD ROM discs readable by processors or computer I/O attachments; information stored on writable storage media, i.e. floppy discs and harddrives; or information convey to a  
15 computer/processor through communication media such as network and/or Internet and/or telephone networks via modems or other interface devices. It should be understood that such media, when carrying processor readable instructions implementing the inventive concept represent  
20 alternate embodiments of the present invention.

#### **BRIEF DESCRIPTION OF DRAWING**

25 The best mode of carrying out the invention as well as preferred embodiments and examples will be explained with reference to the drawing, in which:

Fig. 1 shows a network topology of an ad-hoc

network to which the reconfiguration process according to the present invention is applicable;

- 5    Fig. 2            shows a schematic diagram of a network node according to the present invention;
- Fig. 3            shows a flowchart of operation for the network node shown in Fig. 2;
- 10           Fig. 4            shows a schematic diagram of the software reconfiguration unit shown in Fig. 2;
- Fig. 5            shows a flowchart of operation for different sub-units in the software reconfiguration unit shown in Fig. 4;
- 15           Fig. 6            shows a flowchart of operation for the reconfiguration commitment unit shown in Fig. 2;
- 20           Fig. 7            shows a first example of the reconfiguration process according to the present invention; and
- 25           Fig. 8            shows a second example of the reconfiguration process according to the present invention.

30

#### DESCRIPTION OF BEST MODE AND PREFERRED EMBODIMENTS



In the following, the best mode of carrying out the present invention and preferred embodiments thereof will be explained with respect to the drawing. Throughout the  
5 specification, those parts being identical will be referred to using the same reference numerals.

Fig. 1 shows a network topology of an ad-hoc network to which the reconfiguration process according to the present  
10 invention is applicable. As outlined above, an ad-hoc network is a network formed without any central administration and consisting of network nodes, e.g., mobile devices or stationary devices, that use wireless interfaces to exchange circuit- or packet-switched data. It  
15 should be understood that there is not imposed any restriction either on the type of network node nor on the type of communication in the ad-hoc network according to the present invention.

20 Examples of network nodes are mobile devices like mobile telephones, PDAs, mp3-players, laptop computers, etc. or stationary devices like personal computers. Further, the type of ad-hoc networking may be suitably selected, e.g., on the basis of Bluetooth, HiperLAN/2, personal area  
25 network PAN, IEEE 802.11, WLL Multi-hop Access Systems, according to the MANET initiative taken by the Internet Engineering Task Force IETF.

The same also applies when the ad-hoc network is combined  
30 with a mobile cellular communication environment being

operated on the basis of, e.g., GSM, PDC, GPRS, UMTS, IMT2000, PHS, IS-95, or any hybrid combination thereof.

- Still further, it should be noted that the reconfiguration  
5 process explained in the following is suitable for any type  
of software, such as application software, communication  
software, firmware, operation system software, and/or  
related control parameters.
- 10 Typical examples - without any binding effect on the  
present invention - would be mobile Internet application,  
mobile multi-media application, personalized service  
application, global mobility support, or any other type of  
video, still image, audio data, text data, or speech-based  
15 application supporting software.

- Further, communication-related software may be related,  
e.g., to speech coding, source coding, channel coding,  
radio control according to any standard like WCDMA, TDMA,  
20 FDMA, etc. Further, communication software may be related  
to any ad-hoc network routing protocol, e.g., a distance  
vector routing protocol DSVD, an ad-hoc on-demand distance  
vector routing protocol AODV, and/or a dynamic source  
routing protocol. Still further, examples of firmware could  
25 be a software which is necessary to run a signal processing  
unit, e.g., a DSP, which is usually available within a  
mobile device.

- Fig. 2 shows a schematic diagram of a network node  
30 according to the present invention.

As shown in Fig. 2, the network node 10 comprises a software reconfiguration unit 12 and a reconfiguration commitment unit 14. Further, for exchange of data in the ad-hoc network there is provided a communication unit 16,  
5 and a memory unit 18 allows the storage of any data involved with the operation of the network node 10.

Fig. 3 shows a flowchart of operation for the network node 10 shown in Fig. 2. Operatively, in a step S10, the  
10 transition from an initial software configuration to a target software configuration is prepared, and in a step S12 there is taken a decision to commit to the new software configuration in view of a related reconfiguration result indicated through at least one further network node of the  
15 ad-hoc network.

Generally, the reconfiguration process is therefore split into two phases, a first one to prepare reconfiguration, and a second one to evaluate the reconfiguration result, to  
20 decide on commitment in the new software configuration, and finally to actually commit to the new software or to fall back to the initial software configuration. In the following, further aspects underlying this basis concept of a two-phase reconfiguration process will be discussed in  
25 more detail.

Therefore, according to the present invention there is proposed a network node and method of reconfiguration thereof that ensure consistent software update. This means  
30 that all network nodes have the same consistent software version .

Within the general framework for a reconfiguration process as outlined above with respect to Figs. 2 and 3, the present invention may be varied in a plurality of ways, as  
5 will be outlined in the following.

According to the present invention it is also proposed to determine network nodes in the ad-hoc network executing the reconfiguration at the start of reconfiguration. This is  
10 achieved on the basis of at least one criterion:  
communication capability of a network node; node connectivity; profile data of network node; movement pattern for a mobile network node, i.e. mobile device; hardware status of network node; priority of network node;  
15 and/or group membership of network node.

Communication capability may be related to any specific type of communication - i.e., circuit-switched or packet-switched. Here, one might consider an application scenario  
20 where software being related to packet-switched communication is exchanged only, so that circuit-switched functionality remains unaffected. Therefore, network nodes supporting circuit-switched communication only are not affected by the reconfiguration.

25

In addition, network connectivity in the sense of the present invention implies that only network nodes being integrated into the ad-hoc network should be considered for reconfiguration. Optionally, profile data of network nodes  
30 is integrated into the decision of whether a network node will be reconfigured or not, which profile data may

describe the type of applications, type of communication services, user preferences, etc.

Still further, the movement pattern as criterion for  
5 reconfiguration allows to drop out those mobile network nodes from the reconfiguration process which are in the process of leaving the ad-hoc network anyway. Therefore, an update of such a mobile network node with respect to specific services in the current ad-hoc network becomes  
10 obsolete and should be avoided.

Still further, hardware status of the network node is related both to a dynamic hardware status or a static hardware status. A first example for a dynamic hardware  
15 status is the energy being available within the network node, e.g., the load state of a mobile device battery. Only if enough energy is available within the network node, the reconfiguration of such a network node is meaningful. Further, only if enough memory for a specific type of  
20 application software is available within the network node - as example for a static hardware status - the network node should be reconfigured accordingly.

Still further, additional criteria for determination of  
25 reconfiguration are priority of a network node and group membership. Here, priority assignment to a network node allows to exclude specific network nodes from reconfiguration and to have end users receiving privileged services. Group membership allows to set up different  
30 groups of network nodes and related end users forming ad-hoc networks and to support organizational aspects.

A further aspect of preparation of a transition from an initial software configuration to a target software configuration is related retrieval of software.

5

One option supported by the present invention is to download software to one network node being reachable in the ad-hoc network, and then to distribute the software to each network node participating in the reconfiguration.

10

Remote retrieval of software - e.g., from the Internet - may be achieved via any type of mobile communication environment, e.g., a mobile communication network, a wireless local area network, a personal area network, wireless infrared communication, and Bluetooth communication. Mobile communication may be executed according to any type of standard, i.e., IMT2000, GSM, PDC, PHS, IS-95.

15

20 According to the present invention, the reconfiguration supports also heterogeneous software update for different network nodes or, in other words, software reconfiguration may be achieved in a network node specific way. A further option would be to distribute software reconfiguration to at least two network nodes before subsequent distribution within the ad-hoc network.

25

Yet another option for retrieving of software for reconfiguration would be to use a portable electronic device IC/USIM in a local manner, such that the software necessary for reconfiguration is locally retrieved at

30

each network node. A modification of this approach would be that the portable electronic device is not only carrying software for a specific network node, but for a plurality of network nodes, and that the network node accommodating the portable electronic device is used for distribution of the software to further network nodes.

Besides software retrieval via a mobile communication environment or a portable electronic device, a further option would be that software necessary for reconfiguration is already pre-installed at a network node and therefore is only to be activated to achieve reconfiguration.

While above for selection of network nodes criteria and software retrieval mechanisms have explained as important aspects of the first step S10 to prepare a transition from an initial software configuration to a target software configuration, in the following, important aspects of the decision on commitment to a new software configuration according to step S12 shown in Fig. 3 will be explained.

The present invention relies on the understanding that the reconfiguration to the target software as such is not enough to provide consistency of functionality of network nodes after reconfiguration. Therefore, in addition to the step of software update as first phase of reconfiguration there is provided a second phase for evaluation of the reconfiguration result(s). The evaluation is a prerequisite to decide on commitment to the target software configuration, and for actual transfer to the new software

configuration or establishment of a fallback process in view of the decision result.

In other words, only if a successful reconfiguration is notified within a negotiable maximum time for reconfiguration - indication of reconfiguration result, and executing a fallback to the initial software configuration - through different network nodes participating in the reconfiguration process, there will be achieved commitment to target software configuration(s). Otherwise, there will be initiated a fallback to the starting point of reconfiguration to re-establish communication also when the reconfiguration was not successful. This fallback mechanism allows to maintain operability of the different network nodes in the latter case.

Further, the achievement of a reconfiguration process over different network nodes without central control is achieved through exchange of indication of reconfiguration results between network nodes in the ad-hoc network as will be explained in more detail in the following.

When communication is interrupted during preparation of transition to the target software configuration a first option is to await the installation of the target software configuration and to then use the new communication software for configuration result indication. Alternatively - i.e., when reconfiguration is not successful - a fallback to the initial software configuration could be executed after the time period reserved for reconfiguration, and then the previous initial software configuration and



related communication software may be used for reconfiguration result indication.

From the viewpoint of mobile device operating autonomously,  
5 any mismatch between the communication software after reconfiguration will be an indication to non-successful reconfiguration. To the contrary, a suitable exchange of data also after reconfiguration may be considered as implicit indication of successful reconfiguration without  
10 exchange of related dedicated signaling messages.

Further communication may also continue during the reconfiguration process, typically when no communication software is involved in the reconfiguration process. Here,  
15 the indication of reconfiguration results is easier and may be executed immediately after termination of the preparation phase at each network node. Therefore, at the end of the time period being available for configuration preparation, the decision of commitment to the new software  
20 configuration and/or fallback to the initial software configuration may be taken without any delay.

Optionally, the decision may be taken already before end of the maximum reconfiguration time period when positive  
25 reconfiguration indications are received from all involved network nodes before end of the maximum reconfiguration time period.

According to yet another case, a network node may not  
30 receive any indication of a reconfiguration result at all, irrespective of whether communication is interrupted during

the reconfiguration process or continuously available. In that case, where a network node is completely "blind", the appropriate way is to execute fallback to initial software configuration of the network node before continuation of operation for this network node.

While above general aspects of the present invention have been explained with respect to Figs. 2 and 3, further details thereof will be explained with reference to Figs. 4 to 6, respectively.

Fig. 4 shows a schematic diagram of the software reconfiguration unit shown in Fig. 2.

As shown in Fig. 4, the software reconfiguration unit 12 comprises an ad-hoc network determination unit 20, a software retrieval unit 22, a negotiation unit 24, a coordination unit 26, and a timer unit 28, respectively.

Fig. 5 shows a flowchart of operation of the different sub-units of the software reconfiguration unit 12 as shown in Fig. 4. The steps shown in Fig. 5 are basically related to the time period reserved for preparing the transition from the initial software configuration to the target software configuration.

As shown in Fig. 5, at the beginning of this time period, the ad-hoc network determination unit 20 may determine network nodes in the ad-hoc network which execute a reconfiguration process in step S20 as outlined above. Then, in a step S22 the software retrieval unit will

retrieve software according to the target software configuration as outlined above, and optionally also retrieve related software control parameters, e.g., parameters controlling the operation system of a network  
5 node.

In a step S24, the negotiation unit may negotiate a maximum reconfiguration time period for the network nodes participating in the reconfiguration process. It should be  
10 noted that the maximum reconfiguration time period covers the time period provided for preparation of a transition to the target software state and, optionally a time period for evaluation of reconfiguration results, deciding on commitment on the target software configuration and  
15 executing the commitment or a fallback to the initial software configuration according to the determination result.

Then, in a step S26, the coordination unit 28 may  
20 coordinate start of reconfiguration, when several network nodes execute reconfiguration such that different network nodes execute reconfiguration over the same time period or, in other words, in a coordinated and/or synchronized way. This is supported through provision of a timer unit 28  
25 which operatively starts a reconfiguration timer to measure the actual reconfiguration time against the negotiated maximum reconfiguration time in step S28.

As shown in Fig. 5, a further step S30 - executed by the  
30 software retrieval and transition unit 22 - achieves execution of transition to the new software configuration.

Step S30 divides into the sub-steps installation of new software, indication of success of installation, and forwarding of received reconfiguration results. It should be noted that this transition is achieved autonomously in each single network node and that the coordination is achieved through the reception and forwarding of reconfiguration results from further network node.

While each network node is operating autonomously, it is the continuous reception and forwarding of reconfiguration-related information that allows to distribute reconfiguration status information in the ad-hoc network, which then may again be processed autonomously in each mobile device under the ad-hoc networking paradigm.

As shown in Fig. 5, the ad-hoc determination unit 20 may again be activated after reconfiguration to execute step S32 for determination of mobile devices which are reachable from a reconfigured network node after installation of new software and transition to the target software configuration. This step S32 is particularly useful when network nodes are mobile and roam during the reconfiguration process, so that the ad-hoc network topology changes.

Fig. 6 shows a flowchart of operation for the reconfiguration commitment unit 14 shown in Fig. 2. The operation outlined in this flowchart is related to the operation after the first phase to prepare the transition to the target software configuration or, in other words, after expiration of the timer provided in each network node

to compare the actual reconfiguration time with the negotiated maximum negotiation time.

As shown in Fig. 6, after expiration of the time period reserved for preparing the transition to the new software configuration, the reconfiguration commitment unit 14 will evaluate whether a reconfiguration result was received at all in a step S40. If this is not the case, the reconfiguration commitment unit 14 will decide on fallback to initial software configuration in a step S42.

Otherwise, the reconfiguration commitment unit 14 will determine whether at least one negative indication for a reconfiguration was generated during the reconfiguration. If this is true, the reconfirmation commitment will decide on fallback to initial software configuration according to step S42. Otherwise, a decision to commit to the target software configuration will be taken in a step S46.

The logic behind the decision mechanism shown in Fig. 6 is that the interrogation whether no reconfiguration was received in step S40 corresponds to a situation where a network node can no longer form part of the ad-hoc network, e.g., due to roaming and moving away of a mobile network node from the area of the ad-hoc network. Here, a commitment to a target software configuration which might only be suitable for network nodes in the ad-hoc network would no longer make sense.

Further, the reason to decide on fallback when at least one negative indication of reconfiguration is indicated is to

guarantee interoperability and consistency also after reconfiguration.

However, in an alternative manner and depending on the  
5 number of network nodes participating in the reconfiguration and further the type of reconfiguration, one could also consider a modification of the scheme shown in Fig. 6.

10 E.g., a statistical evaluation of the indicated reconfiguration results using thresholds for acceptable negative reconfiguration level is as well covered by the present invention.

15 It should also be noted that the ad-hoc network may split into several groups, each of which will have a common state of software version, but not necessarily the same. E.g., in a first group, a network node may have to fall back, and therefore the complete group will execute a fallback to the  
20 initial software configuration. To the contrary, the other group may achieve successful reconfiguration for all related network nodes.

Fig. 7 shows a first example of the application of the  
25 reconfiguration process according to the present invention.

For the example shown in Fig. 7 it is assumed that network nodes are mobile devices participating in the reconfiguration process. The mobile devices are not roaming  
30 during reconfiguration. Further, it is assumed that the battery of the mobile device T7 is too low, and that

therefore this mobile device will not participate in the reconfiguration process.

As shown in Fig. 7, mobile device T1 distributes the  
5 software according to the target software configuration for each of the mobile devices T2 to T6. Then, the reconfiguration for all mobile devices takes place, except for mobile device T7.

10 At the end of the time period  $t$  reserved for reconfiguration, the result of reconfiguration is as shown in the lower part of Fig. 7. While mobile devices T3, T1, T4 to T6 execute the reconfiguration process successfully, a failure occurs at mobile device T2 subsequent to  
15 successful reconfiguration of the other mobile devices T1, T3 to T6. The failure is indicated to mobile devices T1, T3 to T6 using failure signaling, e.g., a failure signal or message. At point of time  $t_0+t$ , all participating mobile devices T1 to T6 will decide on fallback to the initial  
20 software version due to failure of the reconfiguration process at mobile devices T2. Alternatively, the same decision may be taken already at receipt of the failure indication from mobile device T2.

25 Fig. 8 shows a second example of the application of the reconfiguration process according to the present invention. For the example shown in Fig. 8 it is assumed that different network nodes are mobile devices that may roam during the reconfiguration process.

As shown in Fig. 8, the reconfiguration process for mobile devices T2, T3, and T6 is executed successfully, while the reconfiguration process at mobile device T4 fails. In addition, shortly before end of the time period reserved  
5 for preparing the transition to the target software configuration, connectivity between mobile devices T2 and T4 is lost, splitting the initial ad-hoc network T1 to T6 into partial ad-hoc networks, T1, T2, T5, and T3, T4, and T6.

10

The reconfiguration process is successful for the first partial ad-hoc network T1, T2, T5, and therefore the first partial ad-hoc network will commit to the target software configuration. To the contrary, the second partial ad-hoc  
15 network will execute a fallback to the initial software configuration due to failure of reconfiguration process in mobile device T4.

While in Figs. 7 and 8 examples have been shown where  
20 communication is possible during the first time period of the reconfiguration for decision on commitment to the new software, an alternative would be that communication is fully interrupted during the reconfiguration process. In the latter case, the reconfiguration result indication  
25 related exchange of information would be postponed until expiry of the time period  $t$  reserved for software reconfiguration.

While above the best mode of carrying out the present  
30 invention as well as further preferred embodiment thereof has been explained with respect to the drawing, it should



be emphasized that all related features explained with reference to the drawing may as well be combined differently to achieve additional variations and modifications of the present invention.